



(12)发明专利申请

(10)申请公布号 CN 110168539 A

(43)申请公布日 2019.08.23

(21)申请号 201880006768.6

(74)专利代理机构 永新专利商标代理有限公司
72002

(22)申请日 2018.01.05

代理人 张立达

(30)优先权数据

62/445,861 2017.01.13 US

15/594,041 2017.05.12 US

(51)Int.Cl.

G06F 16/958(2019.01)

H04L 29/08(2006.01)

(85)PCT国际申请进入国家阶段日

2019.07.12

(86)PCT国际申请的申请数据

PCT/US2018/012441 2018.01.05

(87)PCT国际申请的公布数据

W02018/132303 EN 2018.07.19

(71)申请人 微软技术许可有限责任公司

地址 美国华盛顿州

(72)发明人 D·米伦 K·E·利姆

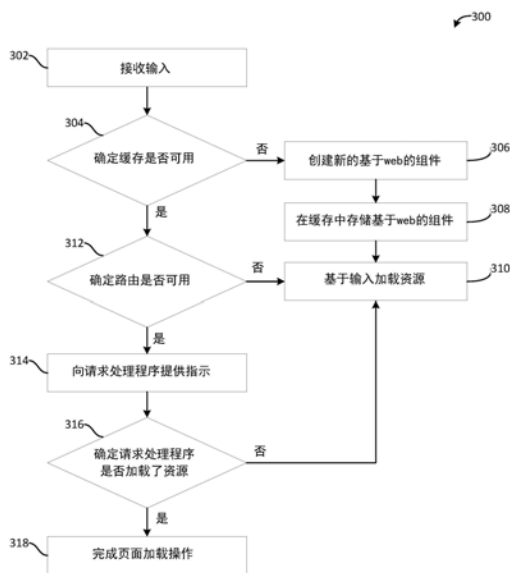
权利要求书2页 说明书14页 附图11页

(54)发明名称

混合应用中的快速页面加载

(57)摘要

本公开的示例涉及用于在混合应用中提供快速页面加载的系统和方法。在示例中,混合应用可以包括本机组件和基于web的组件。为了改进性能,该混合应用可以缓存基于web的组件,该基于web的组件可以进而缓存模板页面。该模板页面可以包括web资源并且可以使用原位更新而被更新,以使得可以在不执行完整页面重载的情况下更新内容。该模板页面可以包括请求处理程序组件,其可以接收针对web资源的请求,确定该web资源可以在模板页面内被加载,以及请求该web资源。接着,该请求处理程序组件可以基于所接收的响应利用内容来更新模板页面。如果该组件不可用,则该web资源可以在基于web的组件内正常地被加载。



1. 一种系统,包括:
至少一个处理器;以及
存储指令的存储器,所述指令当由所述至少一个处理器执行时,实行一组操作,所述一组操作包括:
在具有请求处理程序的基于web的组件中缓存包括一个或多个web资源的模板页面;
由混合应用来接收针对资源的请求;
由所述请求处理程序来确定针对所述资源的所述请求是否能够使用所述基于web的组件中的所述模板页面被处理;
当确定所述请求能够被处理时:
请求所述资源;
响应于所述请求,接收与所述资源相关联的内容;
使用所接收到的内容来更新所述模板页面;以及
由所述混合应用中的所述基于web的组件,使用由所述基于web的组件所缓存的所述一个或多个web资源中的至少一个web资源来显示经更新的模板页面。
2. 根据权利要求1所述的系统,其中,更新所述模板页面包括基于所接收的内容的至少一部分来执行对所述模板页面的原位更新。
3. 根据权利要求1所述的系统,其中,与所述资源相关联的所述内容包括与所述资源相关联的一个或多个另外的web资源的清单。
4. 一种计算机实现的方法,所述方法包括:
在具有请求处理程序的基于web的组件中缓存与web应用相关联的模板页面,其中,所述模板页面包括一个或多个web资源;
由混合应用来接收针对资源的请求;
由所述请求处理程序来确定所述资源是否与所述web应用相关联以及所述模板页面是否可用于处理所述请求;
基于确定所述资源与所述web应用相关联以及所述请求能够被处理:
请求所述资源;
响应于所述请求来接收与所述资源相关联的内容;
基于所接收的内容来更新所述模板页面;以及
由所述混合应用中的所述基于web的组件,使用由所述基于web的组件所缓存的所述一个或多个web资源中的至少一个web资源来显示经更新的模板页面;以及
基于确定所述请求不能够使用所述模板页面被处理,在所述基于web的组件中加载所请求的资源。
5. 根据权利要求4所述的计算机实现的方法,其中,缓存所述模板页面包括基于所述模板页面来准备文档对象模型。
6. 根据权利要求5所述的计算机实现的方法,其中,更新所述模板页面包括利用所接收的内容的至少一部分来至少更新所述文档对象模型。
7. 一种计算机实现的方法,所述方法包括:
在具有请求处理程序的基于web的组件中缓存包括一个或多个web资源的模板页面;
由混合应用来接收针对资源的请求;

由所述请求处理程序来确定针对所述资源的所述请求能够使用所述基于web的组件中的所述模板页面被处理；

基于确定所述请求能够被处理：

请求所述资源；

响应于所述请求来接收与所述资源相关联的内容；

使用所接收的内容来更新所述模板页面；以及

由所述混合应用中的所述基于web的组件，使用由所述基于web的组件所缓存的所述一个或多个web资源中的至少一个web资源来显示经更新的模板页面。

8. 根据权利要求7所述的计算机实现的方法，其中，确定所述请求能够被处理包括确定所述模板页面和所请求的资源是否与web应用相关联。

9. 根据权利要求8所述的计算机实现的方法，其中，缓存所述模板页面包括基于所述模板页面来准备文档对象模型。

10. 根据权利要求9所述的计算机实现的方法，其中，更新所述模板页面包括利用所接收的内容的至少一部分来至少更新所述文档对象模型。

11. 根据权利要求3所述的系统，其中，更新所述模板页面还包括访问所述一个或多个另外的web资源中的至少一个web资源，并且其中，显示经更新的模板页面还包括使用一个或多个另外的web资源中的至少一个web资源。

12. 根据权利要求1所述的系统，其中，所述一组操作还包括：

当确定所述请求不能够使用所述模板页面被处理时，在所述基于web的组件中加载所请求的资源。

13. 根据权利要求4所述的系统，其中，与所述资源相关联的所述内容包括与所述资源相关联的一个或多个另外的web资源的清单。

14. 根据权利要求13所述的计算机实现的方法，其中，更新所述模板页面还包括访问一个或多个另外的web资源中的至少一个web资源，并且其中，显示经更新的模板页面还包括使用一个或多个另外的web资源中的至少一个web资源。

15. 根据权利要求7所述的计算机实现的方法，其中，与所述资源相关联的所述内容包括与所述资源相关联的一个或多个另外的web资源的清单。

混合应用中的快速页面加载

背景技术

[0001] 混合应用可以包括多个组件,其中,每个组件可以提供应用内的不同功能。由于每个组件所提供的独特特性或益处,可以在混合应用中使用不同组件。作为示例,本机组件可以为了性能、本地硬件和存储访问而被使用,并且用来提供特定于设备的特征。相比之下,可以使用基于web(网络)的组件来降低跨平台研发的成本并且在发布对应用的更新时提供较大的灵活性。然而,与本机组件相比,这样的基于web的组件可能操作地较慢。作为结果,尤其是与由本机组件所提供的功能相比,由混合应用所提供的基于web的功能可能表现得缓慢或无响应。

[0002] 在本文中所公开的方面是关于这些和其他一般考虑来进行的。而且,尽管已经讨论了相对具体的问题,但应当理解的是,所述示例不应当受限于解决在背景技术或该公开的其他地方中所标识的具体问题。

发明内容

[0003] 本公开的示例涉及用于在混合应用中提供快速页面加载的系统和方法。在示例中,混合应用可以包括由多个组件所提供的功能。一个组件可以是本机组件,而另一个则可以基于web的组件。作为示例,每个组件可以提供混合应用内的图形用户界面(GUI)中的用户界面元素。为了改进性能,该混合应用可以缓存基于web的组件,该基于web的组件可以进而缓存模板页面。该模板页面可以由一个或多个web资源组成。此外,该模板页面可以允许在基于web的组件内执行原位(in place)更新,以使得可以在不需要执行对web资源的完整页面加载的情况下将内容添加至该模板页面。

[0004] 该模板页面可以包括请求处理程序(handler)组件,其可以接收针对web资源的请求(例如,从移动应用或者从本机组件)。该请求处理程序组件可以确定:可以使用该模板页面来加载web资源。作为结果,该请求处理程序可以请求web资源,从服务器接收响应,并且更新模板页面以包含所接收的内容。如果该请求处理程序组件或基于web的组件不可用或者没有被缓存,则web资源可以在基于web的组件中被正常加载。

[0005] 提供了该发明内容以用简化形式引入在以下的具体实施方式中进一步描述的概念的选择。该发明内容不旨在标识所要求保护的的主题的关键特征或必要特征,也不旨在用来限制所要求保护的的主题的范围。示例的另外的方面、特征和/或优点将在随后的描述中被部分阐述并且将部分地从该描述中是显而易见的,或者可以通过对本公开的实践而习得。

附图说明

[0006] 参考以下附图对非限制和非穷举的示例进行描述。

[0007] 图1图示了用于在混合应用中执行快速页面加载的示例系统的概况。

[0008] 图2A和2B图示了用于在混合应用中执行快速页面加载的示例系统的概况。

[0009] 图3图示了用于在混合应用中执行快速页面加载的示例方法的概况。

[0010] 图4A和4B图示了用于在混合应用中执行快速页面加载的示例方法的概况。

- [0011] 图5是图示了可以利用其实践本公开的方面的计算设备的示例物理组件的框图。
- [0012] 图6A和6B是可以利用其实践本公开的方面的移动计算设备的简化框图。
- [0013] 图7是可以在其中实践本公开的方面的分布式计算系统的简化框图。
- [0014] 图8图示了用于执行本公开的一个或多个方面的平板计算设备。

具体实施方式

[0015] 在下文中参考形成了其一部分并且示出了具体示例性方面的附图更加完整地描述了公开的各种方面。然而，本公开的不同方面可以以许多不同的形式来实现，并且不应当被理解为限于在本文中所阐述的方面；相反，提供了这些方面以使得该公开将是彻底和完整的，并且将向本领域技术人员完全传达所述方面的范围。多个方面可以被实践为方法、系统、或设备。由此，多个方面可以采取硬件实现、完全软件实现、或者组合了软件和硬件方面的实现的形式。因此，以下的详细描述不被认为是限制性意义。

[0016] 本公开提供了用于在混合应用中执行快速页面加载的系统和方法。混合应用可以包括多个组件，并且每个组件可以使用不同的计算语言被编写，使用不同的框架被生成，或者由不同的过程来执行，等等。作为示例，混合应用可以包括本机组件（例如，以诸如C、C++、Objective-C、Swift等之类的计算语言被编写）以及基于web的组件（例如，以诸如HTML、JavaScript、CSS、PHP等之类的基于web的语言被编写，从web服务器获取，以及在混合应用中被显示）。本领域技术人员将会意识到，多种技术、语言、和框架可以是混合应用的一部分而不脱离该公开的精神。

[0017] 在一些示例中，可以使用包括混合应用的组件来基于对每个组件独特的特性来执行不同任务，所述特性包括但不限于性能考量、研发成本、或者安全性特征。如在上文中所讨论的，混合应用可以包括本机组件和基于web的组件。可以使用本机组件来提供设备集成（例如，访问本地存储的文件，与具体硬件功能交互等）、显示图形用户界面（GUI）元素、以及处理用户输入（例如，处理用户导航、解译语音输入等），等等。相比之下，基于web的组件可以对远程资源（例如，包括图像、视频、样式表、或JavaScript代码等的网页或web应用）进行访问、显示、和交互。由此，可以使用基于web的组件来降低与跨平台研发相关联的研发成本、提供快速部署、并且保持对混合应用的某些方面的控制。

[0018] 不幸的是，尽管有上文所讨论的益处，混合应用可能呈现出某些折衷。在示例中，基于web的组件与本机组件相比可能展现出较慢的执行时间并且响应性较差。作为结果，与由基于web的组件所提供的功能相比，由混合应用的本机组件所提供的功能可能看起来执行得更好或者响应性更好。因此，尽管基于web的组件可以提供降低的研发成本和更大的灵活性，但是由基于web的组件所提供的内容可能响应性较差，这是因为与该内容相关联的资源可能是在远离混合应用的位置处被生成或存储的。在本文中所公开的方面涉及通过以下操作来改进这样的组件的性能特性：缓存组件的至少一部分，检测到该组件被缓存，以及关于当在混合应用中执行时降低组件的开销（例如，处理时间、远程资源请求等）的方式使用所缓存的组件。

[0019] 混合应用可以缓存一个或多个组件以便降低混合应用和/或其他组件所需的处理时间的量。对组件进行缓存可以包括在事件（例如，用户交互、计时器超时、接收到指示等）之前或者在为事件做准备时执行一个或多个动作。所述一个或多个动作的结果可以被混合

应用存储(例如,在临时位置处,在存储设备或存储系统上,在系统存储器中,等等),并且在需要时(例如,一旦事件发生,响应于所接收的指示等)被获取。在一些示例中,混合应用可以在该混合应用被启动时缓存一个或多个组件。在另一示例中,混合应用可以在初始动作之后缓存一个或多个组件(例如,响应于第一事件,缓存可以在为潜在的第二个事件做准备时进行)。

[0020] 在其他示例中,组件自身可以执行缓存(例如,对一个或多个页面特征、远程资源等的缓存)。在一个示例中,诸如网页、图像、JavaScript文档或库等之类的“模板”资源可以被组件加载,其中,该模板资源包括通常由该组件使用以执行操作或者通常由与该模板资源相关联的其他资源所加载的一个或多个其他资源。相关联的资源 and 模板资源可以共享共同的特征,例如共同的设计、布局、库、或框架等。作为结果,对模板资源进行加载可以使得其他资源和特征在接收到针对相关联的资源请求之前被缓存。例如,一些资源可以是特定于租户的或者具有特定于租户的共享的页面特征,例如共同的框架。由此,可以创建具有共享的页面特征的模板。由此,在加载新的特定于租户的资源时,仅需要将不同于共享的页面特征的内容加载到该模板中,这减少了向用户呈现页面所需的时间。

[0021] 作为示例,基于web的组件可以加载模板页面,这使得一个或多个资源(包括但不限于JavaScript库、图像、或文档)在为可能要求或依赖于一个或多个所下载的资源web资源请求做准备时被下载和存储。作为加载模板页面的结果,空白的HTML文档对象模型(DOM)可供使用,其中,框架或库可以已经被编译,样式表可以已经被加载和评估,并且图像可以已经被缓存,等等。与模板页面相关联的web页面可以使用该空白DOM来加载,例如可以通过更新DOM以包含来自相关联的页面的内容和其他信息来执行原位更新。在一些示例中,模板页面和相关联的页面可以是特定于租户的资源,这在于它们可以与该租户的网页或web应用相关联。该模板页面和相关联的页面可以由MICROSOFT SHAREPOINT所采用的Modern Page。

[0022] 与传统的混合应用相比,为组件缓存一个或多个资源可以显著地降低组件用来处理请求的处理时间的量。作为示例,在基于web的组件可能花费7秒钟来完成请求时,缓存可以将处理时间减半至大约4秒或更少。另外地,在基于web的组件中缓存模板页面以便执行针对所请求的资源的原位更新可以将大约4秒的处理时间进一步降低至小于1秒。因此,与传统的混合应用相比,在本文中所描述的方面可以提供至少7倍的性能增加。

[0023] 诸如本机组件、基于web的组件、和请求处理程序组件之类的构成混合应用的多个组件可以彼此交互,以使得它们提供无缝的用户体验。在一个示例中,组件可以交换状态信息和其他数据。在另一个示例中,一个组件能够激活另一个组件,以使得所述另一个组件可以作为所提供的指示的结果而在混合应用的GUI中被视觉地呈现。本领域技术人员将意识到,可以由混合应用来提供组件之间的任何交互而不脱离该公开的精神。

[0024] 多个组件可以使用多种机制来与彼此或与用户交互,所述多种机制包括但不限于:应用编程接口(API)、统一资源标识符(URI)或统一资源定位符(URL)回调方案、或者经由HTML5Window.postMessage()函数调用。在一些示例中,所缓存的组件可用于其他组件,以使得即使所缓存组件还没有被混合应用或其组件所调用,其他组件也能够与所缓存的组件进行交互。组件可以与所缓存的组件进行交互以确定其是否可用于处理或者确定什么资源被加载或可用,等等。类似地,所缓存的组件可以向混合应用或其他组件提供有关其可用

性或能力等的指示。

[0025] 图1图示了用于在混合应用中执行快速页面加载的示例系统100的概况。系统100由客户端102和服务器104组成。在示例中,客户端102和服务器104中的每个可以是计算设备,包括但不限于:个人计算设备、平板计算设备、移动计算设备、或分布式计算设备。在一些示例中,服务器104可以是分布式计算环境中的租户。客户端102可以与服务器104进行通信(例如,使用无线或有线网络)。

[0026] 服务器104可以包括页面生成组件106。页面生成组件106可以响应于由服务器104所接收到的访问请求来生成网页。在示例中,服务器104可以使用页面生成组件106以向一个或多个客户端(例如,客户端102)提供web应用或web服务(例如,MIRCOSOF T SHAREPOINT、MIRCOSOF T OFFICE 365、MIRCOSOF T OUTLOOK、GOOGLE GMAIL等)。在一些示例中,可以使用页面生成组件106来为共享的托管环境中的多个租户生成网页。在其他示例中,可以存在多个页面生成组件,其中,每个页面生成组件可以为共享的托管环境的一个或多个租户生成网页。

[0027] 客户端102可以包括混合应用108。混合应用108可以至少部分是在客户端102上执行的本地应用。混合应用可以包括本机组件110、基于web的组件112、和请求处理程序组件114。请求处理程序组件114在本文中也可以被称为“路由器”组件。在一些示例中,基于web的组件112可以包括请求处理程序组件114。本机组件110可以是客户端102能够本地地执行(例如,在没有虚拟机或语言解译器等的情况下)的组件。本机组件110可以提供多种功能,包括但不限于:生成或渲染GUI元素,与客户端102的硬件组件进行通信,以及访问本地存储在客户端102上的文件。在一些示例中,本机组件110可以发出针对由服务器104所提供(例如,使用页面生成组件106)的资源的一个或多个请求,所述资源可以由混合应用108使用基于web的组件112在客户端102上显示。

[0028] 基于web的组件112可以访问远程资源(例如,服务器104使用页面生成组件106所提供的web资源)。基于web的组件112可以提供一个或多个元素以便在混合应用108的GUI内显示,以使得用户可以与由基于web的组件112所显示的元素进行交互。在示例中,基于web的组件112可以接收来自用户的输入,并且可以基于所接收到的输入来执行本地操作(例如,使用由基于web的组件112所执行的脚本语言)、向服务器104发送数据、或者向混合应用108的另一组件(例如,本机组件110或请求处理程序组件114)提供指示,等等。在一些示例中,本机组件110可以执行操作并且在基于web的组件112旁显示GUI元素,以使得每个组件都可以被混合应用108同时使用。作为示例,本机组件110可以为混合应用108提供页眉、页脚、和导航,而基于web的组件112可以从本机组件110接收信息并且在混合应用108内显示内容。在一些示例中,基于web的组件112可以从本机组件110或请求处理程序组件114接收信息,并且反之亦然。

[0029] 请求处理程序组件114可以与混合应用108以及与组件110-112进行通信。在一些示例中,请求处理程序组件114可以被包括在诸如基于web的组件112之类的组件中或者由该组件来实现。请求处理程序组件114可以从本机组件110或基于web的组件112截取或接收指示,该指示包括但不限于针对资源(例如,web资源、文档等)的请求或者状态更新请求。该指示可以使用API、回调、或其他方法来提供。基于该指示,请求处理程序组件114可以确定web请求是可以由基于web的组件112来处理,如果可以,则向基于web的组件112提供内容,

基于web的组件112可以用于加载和显示与所请求的web资源相关的内容。在一些示例中,请求处理程序组件114可以实现这样的协议,该协议可以由其他组件使用来与请求处理程序组件114通信。

[0030] 作为示例,请求处理程序组件114可以使用“router://”协议来针对指示进行收听,“router://”可以后跟有资源或对资源的引用(例如,使用URL、URI等)等。指示可以使用API、URL、或URI方案等被提供。接着,请求处理程序组件114可以评估该指示以确定该指示是否应当被路由或者应当执行什么动作。评估该指示可以包括访问资源的至少一部分(例如,报头、资源的前一半等)或者与资源相关联的元数据。在另一个示例中,可以基于模式匹配来评估该指示,其中,将所提供的数据与一个或多个模式匹配规则进行比较以确定应当采取什么动作。本领域技术人员将会意识到,可以使用任何数量的技术或不同的协议来评估该指示并且确定随后的动作。

[0031] 在一些示例中,混合应用108可以在该混合应用108首次被客户端102加载时缓存基于web的组件112。缓存基于web的组件112可以包括初始化基于web的组件112,以使得基于web的组件112可以访问一个或多个web资源(例如,如可以由服务器104和页面生成组件106所提供的)并且存储它们以供未来使用。在一些示例中,基于web的组件112可以访问模板页面,该模板页面可以是诸如Modern Page之类的特定于租户的web资源。该模板页面可以包括网页或web应用所要求的一个或多个资源,但是可能还没有加载内容。在一些示例中,请求处理程序组件114可以知晓一个或多个组件的状态(例如,组件是否被加载或被缓存,基于web的组件112是否已经加载了一个或多个资源,等等)。请求处理程序组件114可以作为基于web的组件112加载空白页面的结果而被加载,并且请求处理程序组件114可以之后确定基于web的组件112是否可用于执行原位更新以处理web请求。

[0032] 当被客户端102加载时,混合应用108可以显示GUI。该GUI可以包括能够由本机组件110所提供的元素,其包括但不限于:页眉、页脚、和导航窗格。在一些示例中,本机组件110还可以在正文元素中显示内容。所显示的元素可以包括一个或多个引用,当所述引用被用户选择时或者响应于另一个事件,可以调用混合应用108的不同组件(例如,基于web的组件112和/或请求处理程序组件114)。

[0033] 当这样的元素被选择时,可以确定基于web的组件112是否可用于处理该指示。在一些示例中,基于web的组件112先前可能没有被缓存或者可能不再可用。作为结果,基于web的组件112可以被加载或者用于处理该指示。类似地,如果请求处理程序组件114不可用但是基于web的组件被加载,则基于web的组件112可以用于处理该指示。处理该指示可以包括获取并显示该指示的一个或多个资源。

[0034] 在一些示例中,可以确定请求处理程序组件114可用于处理该指示。作为结果,请求处理程序组件114可以接收指示(例如,使用API、协议等)并且确定该指示是否可以被路由(例如,如在本文中所描述的,使用模式匹配或其他分析)。在示例中,请求处理程序组件114可以确定该指示包括针对与先前加载的模板页面相关联的web资源的请求。该模板页面和所请求的web资源可以是web应用的一部分(例如,它们中的每个都可以是MIRCOSFT SHAREPOINT内的Modern Page或者是可以针对其创建模板的其他资源)。在另一个示例中,请求处理程序组件114还可以评估基于web的组件112的状态。如果请求处理程序组件114确定该指示可以由基于web的组件112来处理(例如,通过原位更新),则其可以向基于web的组

件112提供指示。

[0035] 在一个示例中,请求处理程序组件114可以获取与指示相关联的内容并且将该内容提供至基于web的组件112以便加载。另外,基于web的组件112可以使模板页面被加载,以使得可以通过利用删减的内容(例如,其可以排除页眉、页脚、导航窗格等中的至少一个)更新模板页面(例如,如上文所描述的,缓存的空白DOM)来处理该内容。作为结果,当基于web的组件112从请求处理程序组件114接收到内容时,其可以执行对GUI的原位更新以加载与所指示的web资源相关联的内容,同时将本地缓存的资源用于web应用的其他方面。这可以降低与使用基于web的组件112执行正常页面加载相关联的计算开销。在一些示例中,尽管基于web的组件112可能没有接收到与web资源相关联的某些界面元素,但是基于web的组件112仍然可以更新或以其他方式修改GUI的其他元素(例如,通过改变在标题栏中所显示的URL,相应地更新导航窗格,等等)。

[0036] 在一些示例中,服务器104或页面生成组件106可以在从请求处理程序组件114接收到针对web资源的请求时检测到有可能发送包括该web资源的删减的表示的响应,其中,该删减的表示可以省略了页眉、页脚、或导航窗格等。在示例中,该删减的表示可以以常规的未删减的页面的计算语言(HTML、JSON、XML等)之外的格式来提供。另外地,该删减的表示可以包括另外的信息,其包括但不限于可以在显示web资源时被加载的另外的资源的清单(例如,另外的JavaScript库、图像资源等)或者所述另外的资源本身。接着,基于web的组件112或请求处理程序组件114可以使用该信息来加载先前可能没有被缓存的另外的资源。

[0037] 图2A图示了用于在混合应用中执行快速页面加载的示例系统200的概况。系统200可以由页眉202、页脚204、和正文206组成。系统200可以被显示在计算设备上,例如,图1中的客户端102。系统200可以是混合应用,其中,页眉202、页脚204、和正文206中的至少一个是由本机组件(例如,图1中的本机组件110)所提供的GUI元素。在示例中,页眉202和页脚204中的每个可以提供导航元素(例如,标题栏、URL栏等)、信息显示、或者其他视觉元素。正文206可以由本机组件使用以向用户显示内容或信息,所述内容或信息包括但不限于:文本数据、图像数据、视频数据、或者计算机生成的渲染。本机组件可以基于事件(例如,从用户接收到的交互,计时器的超时、从另一组件或计算设备接收到的指示,等等)来更新对页眉202、页脚204、或正文206的显示。

[0038] 图2B图示了其中激活了基于web的组件的示例系统200的概况。基于web的组件(例如,基于web的组件114)可以已经响应于从混合应用的本机组件所接收到的指示而被激活。类似于图2A,本机组件可以提供页眉202和页脚204。导航窗格208也可以由本机组件来提供。然而,正文210可以由基于web的组件来提供,其中,一个或多个web资源可以在混合应用的GUI内被获取和显示。基于web的组件可以与本机组件交互以便更新页眉202、页脚204、或导航窗格208的至少一部分。作为示例,当不同的web资源由基于web的组件使用正文210来显示时,导航元素(例如,向后和向前按钮、URL栏、页面标题等)或导航窗格208可以相应地被更新。

[0039] 图3图示了用于在混合应用中执行快速页面加载的示例方法300的概况。方法300可以由计算设备来执行,例如图1中的客户端102。在一些示例中,方法300可以由混合应用来执行,例如图1中的混合应用108。方法300在操作302处开始,在302处,可以接收输入。该输入可以是来自混合应用的组件所接收的指示,所述组件例如是图1中的本机组件110。该指

示可以包括针对web资源的请求。

[0040] 移动至操作304,可以关于缓存是否可用来进行确定。该确定可以包括评估所缓存的组件(例如,图1中的基于web的组件112)是否可用于处理所接收到的输入。如果确定该缓存不可用,则流程分支向N0(否)以去往操作306,其中,可以创建新的基于web的组件。创建基于web的组件可以包括在基于web的组件内加载和缓存模板页面或其他web资源。在操作308处,新的基于web的组件可以被存储在缓存中。基于web的组件可以在存储设备上或者使用任何其他缓存技术而被缓存在系统存储器中。在操作310处,web资源可以被加载于基于web的组件中。加载web资源可以包括与远程服务器(例如,图1中的服务器104)或其他计算设备进行通信。在示例中,多个其他web资源可以作为加载web资源的结果而被加载。流程在操作310处终止。

[0041] 然而,如果确定缓存可用,则流程分支向YES(是)以去往操作312,其中,关于路由是否可用来进行确定。该确定可以包括:评估系统存储器中是否存在请求处理程序(例如,图1的请求处理程序组件114),评估全局变量或函数调用是否可用,等等。如果确定路由不可用,则流程分支向N0以去往操作310,其中,资源可以如上文所讨论的基于所接收的输入而被加载。流程在操作310处终止。

[0042] 然而,如果确定路由可用,则流程分支向YES(是)以去往操作314,在314处,可以向请求处理程序提供指示。该指示可以使用API、URL、或者URI方案或协议等来提供。响应于接收到该指示,请求处理程序可以根据在本文中所公开的方面来评估所接收到的指示(例如,使用模式匹配,访问与该指示相关联的资源的至少一部分,等等)。接着,请求处理程序可以确定资源是否可以在缓存的基于web的组件中被加载。在示例中,该确定可以包括确定所请求的资源与模板页面相关联(例如,所请求资源和模板页面两者都是web应用的部分,例如MIRCOSOFT SHAREPOINT内的Modern Page)。如果缓存的基于web的组件可以用于加载该资源,则请求处理程序可以使用该缓存的基于web的组件来加载该资源。如上文所讨论的,可以从服务器接收对web资源的删减的表示,其接着可以用于执行对在基于web的组件中所加载的模板页面的原位更新。

[0043] 移动至操作316,可以关于请求处理程序是否加载了该资源来进行确定。该确定可以是基于以下动作的:接收到对在操作314处提供至请求处理程序的指示的响应,或者经由如在本文中所公开的API或其他回调机制而从请求处理程序接收到指示。如果请求处理程序没有加载该资源,则流程分支向否(N0)以去往操作310,在310处,可以如上文所讨论的基于所接收的输入来加载该资源。接着,流程在操作310处终止。然而,如果请求处理程序加载了该资源,则流程分支向是(YES)以去往操作318,在318处,页面加载操作完成。在一些示例中,移动应用可以执行一个或多个动作以作为完成页面加载操作的结果,所述动作包括但不限于:更新其GUI的元素、激活由组件所提供的元素、或者收集与输入相关的信息。流程在操作314处终止。

[0044] 图4A图示了用于在混合应用中执行快速页面加载的示例方法的概况。该方法可以由移动应用402、基于web的组件404、请求处理程序组件406、和web服务408来执行。移动应用402可以是在用户设备(例如,客户端102)上执行的混合应用(例如,图1中的混合应用108),其中,移动应用402的本机组件(未示出)可以与基于web的组件404交互。基于web的组件404可以是由移动应用402用来显示一个或多个资源(例如,如由web服务408所提供的)的

基于web的组件(例如,图1中的基于web的组件112)。移动应用402可以使用API、回调方案、或者其他通信机制来与基于web的组件404交互。请求处理程序组件406(例如,图1中的请求处理程序组件114)可以由基于web的组件404加载,并且用于处理和加载针对web资源的请求。Web服务408可以是能够向基于web的组件404和请求处理程序组件406提供一个或多个网页或web应用的服务器或其他计算设备(例如,图1中的服务器104)。

[0045] 该方法在操作410处开始,在410处,移动应用402可以被启动。在操作412处,移动应用402可以生成对基于web的组件(例如,基于web的组件404)的缓存。该基于web的组件可以被加载以使得一个或多个web资源被该基于web的组件所缓存。在示例中,基于web的组件可以如在本文中所公开的那样加载并缓存模板页面。在另一示例中,基于web的组件404可以加载请求处理程序组件(例如,请求处理程序组件406)。移动至操作414,可以向移动应用402提供关于对基于web的组件404的缓存可用的指示。

[0046] 在操作416处,可以接收针对项目(例如,web或其他远程资源)的请求,这使得指示被提供至基于web的组件404。可以作为用户与移动应用402的交互或其他事件的结果来接收该请求。移动至操作418,可以关于确定页面路由器是否存在来进行确定(例如,请求处理程序组件406)。该确定可以包括评估全局变量,确定系统存储器中是否存在路由器,等等。在操作420处,确定路由器存在。

[0047] 作为结果,可以向请求处理程序组件406提供指示,其中,该指示包括与针对在操作416处所接收的项目的请求相关的信息。该指示可以符合协议(例如,“router://”协议),调用API函数,或者采用回调方案,等等。请求处理程序组件406可以接收该指示,并且可以在操作424处从web服务408请求所指示的项目。Web服务408可以确定该请求是由请求处理程序组件406生成的(例如,由于具体的用户-代理串,随请求提供的GET或POST变量,等等),并且可以将对该项目的缩减的表示提供至请求处理程序组件406。在操作426处,请求处理程序组件406可以接收缩减的内容并且相应地对其进行渲染。在示例中,请求处理程序组件406可以使用所接收的内容来执行对由基于web的组件404所加载的模板页面的原位更新。在一些示例中,请求处理程序组件406还可以更新由基于web的组件404所提供的或者由移动应用402所显示的其他方面,所述其他方面包括但不限于:URL栏、标题栏、或导航元素(例如,向后和向前按钮、导航窗格等)。

[0048] 在操作428处,基于web的组件404可以接收关于该项目是由请求处理程序组件406所加载的指示。作为该指示的结果,基于web的组件404可以在操作430处向移动应用402提供页面加载完成的指示。在一些示例中,移动应用402可以作为接收到该指示的结果而执行一个或多个动作,所述动作包括但不限于:更新GUI的元素,激活由组件(例如,基于web的组件404)所提供的元素,或者收集与该请求相关的信息。流程在操作430处结束。

[0049] 图4B图示了用于在混合应用中执行快速页面加载的示例方法的概况。该方法可以由移动应用402、基于web的组件404、请求处理程序组件406、和web服务408来执行。移动应用402可以是在用户设备(例如,客户端102)上执行的混合应用(例如,图1中的混合应用108),其中,移动应用402的本机组件(未示出)可以与基于web的组件404交互。基于web的组件404可以是由移动应用402用来显示一个或多个资源(例如,由web服务408所提供的)的基于web的组件(例如,图1中的基于web的组件112)。移动应用402可以使用API、回调方案、或者其他通信机制来与基于web的组件404交互。请求处理程序组件406(例如,图1中的请求处

理程序组件114)可以由基于web的组件404加载,并且用于处理和加载针对web资源的请求。Web服务408可以是能够向基于web的组件404和请求处理程序组件406提供一个或多个网页或web应用的服务器或其他计算设备(例如,图1中的服务器104)。

[0050] 该方法在操作410处开始,在410处,移动应用402可以被启动。在操作412处,移动应用402可以生成对基于web的组件(例如,基于web的组件404)的缓存。然而,在操作450处,可以确定没有缓存可用。缓存之所以不可用可能是因为web视图组件不能够加载,或者是因为web视图组件在其缓存中不再具有先前缓存的一个或多个资源(例如,模板页面、所要求的JavaScript库等)。作为结果,当在操作452处接收到针对项目的请求时,基于web的组件404可能不像图4A中的情况下那样使用请求处理程序组件406。替代地,在操作454处,基于web的组件404可以将针对项目的请求直接发送至web服务408。

[0051] 在接收到该请求时,web服务408可以确定该请求是由基于web的组件在不使用请求处理程序组件(例如,由于缺少具体的用户-代理串,随请求提供的GET或POST变量等)的情况下生成的,并且可以将对该项目的完整表示(例如,包含页眉、页脚、或导航窗格等)提供至请求处理程序组件406。在一些示例中,web服务408可以不作出确定并且可以默认地提供对该项目的完整表示。在操作456处,请求处理程序组件406可以作为接收到该内容的结果而被加载。在示例中,请求处理程序组件406可以被包括在所接收的内容(例如,作为JavaScript库、计算框架、或其他计算资源)之内。此外,基于web的组件404可以下载与所接收到的内容相关联的web资源,其包括但不限于图像、脚本资源、和样式表。与在图4A中所示出的流程相反,可以不执行原位更新,这是由于基于web的组件404先前没有加载模板页面。在一些示例中,基于web的组件404和/或请求处理程序组件406可以更新由基于web的组件404所提供或者由移动应用402所显示的GUI的其他方面,所述方面包括但不限于:URL栏、标题栏、或导航元素(例如,向后和向前按钮、导航窗格等)。

[0052] 在操作458处,基于web的组件404可以接收关于项目是由请求处理程序组件406所加载的指示。作为该指示的结果,基于web的组件404可以在操作460处向移动应用402提供页面加载完成的指示。在一些示例中,移动应用402可以作为接收到该指示的结果而执行一个或多个动作,所述动作包括但不限于:更新GUI的元素,激活由组件(例如,基于web的组件404)所提供的元素,或者收集与该请求相关的信息。流程在操作460处结束。

[0053] 图5-8以及相关描述提供了对在其中可以实践本公开的方面的多种操作环境的讨论。然而,关于图5-8所示出并讨论的设备和系统是出于示例和说明的目的,而不对可以用来实践在本文中所描述的公开的方面的大量计算设备配置进行限制。

[0054] 图5是图示了可以利用其实践本公开的方面的计算设备500的物理组件(例如,硬件)的框图。下文所描述的计算设备组件可以适用于上文所描述的计算设备,包括客户端102和服务器104。在基本配置中,计算设备500可以包括至少一个处理单元502和系统存储器504。取决于计算设备的配置和类型,系统存储器504可以包括但不限于易失性存储单元(例如,随机存取存储器)、非易失性存储单元(例如,只读存储器)、闪存存储器、或者这些存储器的任何组合。系统存储器504可以包括操作系统505以及一个或多个程序模块506,所述一个或多个程序模块506适合于执行如在本文中所公开的诸如请求处理程序组件524和基于web的组件526之类的各种方面。例如,操作系统505可以适用于控制计算设备500的操作。此外,本公开的实施例可以结合图形库、其他操作系统、或任何其他应用来实践,并且不限

于任何特定的应用或系统。该基本配置是在图5中由虚线508内的那些组件示出的。计算设备500可以具有另外的特征或功能。例如,计算设备500还可以包括例如磁盘、光盘、或磁带之类的另外的数据存储设备(可移动的和/或不可移动的)。这样另外的存储单元是在图5中由可移动存储设备509和不可移动存储设备510示出的。

[0055] 如上所述,多个程序模块和数据文件可以被存储在系统存储器504中。当在处理单元502上执行时,程序模块506(例如,应用520)可以执行以下过程,其包括但不限于在该本文中所描述方面。可以根据本公开的方面使用的其他程序模块可以包括:电子邮件和联系人应用、文字处理应用、电子表格应用、数据库应用、幻灯片演示应用、绘图或计算机辅助应用程序等。

[0056] 此外,本公开的实施例可以被实现在包括分立电子元件的电子电路中、包含逻辑门的封装或集成电子芯片中、利用微处理器的电路中、或者包含电子元件或微处理器的单个芯片上。例如,可以经由片上系统(SOC)来实践本公开的实施例,其中图5中所示出的每个或许多组件可以被集成到单个集成电路上。这样的SOC设备可以包括一个或多个处理单元、图形单元、通信单元、系统虚拟化单元、以及各种应用功能,其全部作为单个集成电路而被集成(或“烧”)到芯片基底上。当经由SOC进行操作时,可以经由在单个集成电路(芯片)上与计算设备500的其他组件集成的专用逻辑来对在本文中关于客户端切换协议的能力所描述的功能进行操作。还可以使用能够执行逻辑操作(例如AND、OR和NOT)的其他技术来实践本公开的实施例,所述其他技术包括但不限于机械、光学、流体、和量子技术。另外,可以在通用计算机内或在任何其他电路或系统中实践本公开的实施例。

[0057] 计算设备500还可以具有一个或多个输入设备512,例如键盘、鼠标、笔、声音或语音输入设备、触摸或滑动输入设备等。还可以包括诸如显示器、扬声器、打印机等的输出设备514。前述设备是示例并且可以使用其他设备。计算设备500可以包括允许与其他计算设备550进行通信的一个或多个通信连接516。合适的通信连接516的示例包括但不限于射频(RF)发射机、接收机和/或收发机电路;通用串行总线(USB)、并行、和/或串行端口。

[0058] 如在本文中所使用的术语计算机可读介质可以包括计算机存储介质。计算机存储介质可以包括以任何用于存储信息(例如,计算机可读指令、数据结构、或程序模块)的方法或技术来实现的易失性的和非易失性的、可移动的和不可移动的介质。系统存储器504、可移动存储设备509、和不可移动存储设备510都是计算机存储介质的示例(例如,存储器存储单元)。计算机存储介质可以包括RAM、ROM、电可擦只读存储器(EEPROM)、闪存存储器或其他存储器技术、CD-ROM、数字通用盘(DVD)或其他光存储单元、盒式磁带、磁带、磁盘存储器或其他磁存储设备、或者可用于存储信息并且可以由计算设备500访问的任何其他制品。任何这样的计算机存储介质都可以是计算设备500的一部分。计算机存储介质不包括载波或其他经传播或经调制的数据信号。

[0059] 通信介质可以由计算机可读指令、数据结构、程序模块、或经调制的数据信号(例如,载波或其他传播机制)中的其他数据来实施,并且包括任何信息传递介质。术语“经调制的数据信号”可以描述具有以关于将信息编码在信号中的方式设置或改变的一个或多个特性的信号。作为示例而非限制,通信介质可以包括诸如有线网络或直接有线连接之类的有线介质,以及诸如声学的、射频(RF)的、红外的、和其他无线介质之类的无线介质。

[0060] 图6A和6B示出了可以利用其实践本公开的实施例的移动计算设备600,例如移动

电话、智能电话、可穿戴计算机(例如,智能手表)、平板计算机、膝上型计算机等。在一些方面中,客户端可以是移动计算设备。参考图6A,图示了用于实施所述方面的移动计算设备600的一个方面。在基本配置中,移动计算设备600是具有输入部件和输出部件两者的手持式计算机。移动计算设备600通常包括显示器605以及允许用户将信息输入到移动计算设备600中的一个或多个输入按钮610。移动计算设备600的显示器605也可以充当输入设备(例如,触摸屏显示器)。如果被包括,则可选的侧输入部件615允许另外的用户输入。侧输入部件615可以是旋转开关、按钮、或任何其他类型的手动输入元件。在可替代的方面中,移动计算设备600可以结合有更多或更少的输入部件。例如,在一些实施例中,显示器605可以不是触摸屏。在又一个实施例中,移动计算设备600是诸如蜂窝电话之类的便携式电话系统。移动计算设备600还可以包括可选的小键盘635。可选的小键盘635可以是物理小键盘,或者是在触摸屏显示器上生成的“软”小键盘。在各种实施例中,输出元件包括用于示出图形用户界面(GUI)的显示器605、可视指示器620(例如,发光二极管)、和/或音频换能器625(例如,扬声器)。在一些方面中,移动计算设备600包含用于向用户提供触觉反馈的振动换能器。在另一方面中,移动计算设备600包含用于将信号发送到外部设备或者从外部设备接收信号的输入和/或输出端口,例如音频输入(例如,传声器插孔)、音频输出(例如,耳机插孔)、以及视频输出(例如,HDMI端口)。

[0061] 图6B是示出了移动计算设备的一个方面的框架的框图。也就是说,移动计算设备600能够包含用于实现一些方面的系统(即框架)602。在一个实施例中,系统602被实现为能够运行一个或多个应用(例如,浏览器、电子邮件、日历、联系人管理器、消息传送客户端、游戏、以及媒体客户端/播放器)的“智能电话”。在一些方面中,系统602被集成为计算设备,例如集成的个人数字助理(PDA)和无线电话。

[0062] 一个或多个应用程序666可以被加载到存储器662中并且在操作系统664上运行或者与操作系统264相关联地运行。应用程序的示例包括电话拨号程序、电子邮件程序、个人信息管理(PIM)程序、文字处理程序、电子表格程序、互联网浏览器程序、消息传送程序等。系统602还包括存储器662内的非易失性存储区域668。非易失性存储区域668可以用于存储如果系统602断电而不应丢失的持久信息。应用程序666可以在非易失性存储区域668中使用并存储信息,例如电子邮件或者由电子邮件应用使用的其他消息等。同步应用(未示出)也驻留在系统602上,并且被编程为与驻留在主计算机上的对应的同步应用进行交互,从而将存储在非易失性存储区域668中的信息与存储在主计算机上的对应的信息保持同步。应当理解的是,其他应用可以被加载到存储器662中并且在本文所描述的移动计算设备600上运行(例如,搜索引擎、提取器模块、相关性排名模块、回答评分模块等)。

[0063] 系统602具有电源670,其可以被实现为一个或多个电池。电源670还可以包括外部电源,例如AC适配器或给电池供电或充电的加电对接托架。

[0064] 系统602还包括执行发送和接收射频通信的功能的无线电接口层672。无线电接口层672经由通信载波或服务提供商来促进系统602与“外部世界”之间的无线连通性。来往于无线电接口层672的传输是在操作系统664的控制下进行的。换句话说,可以将由无线电接口层672所接收的通信经由操作系统664散播至应用程序666,并且反之亦然。

[0065] 视觉指示器620可以用于提供视觉通知,和/或音频接口674可以用于经由音频换能器625产生可听见的通知。在所示出的实施例中,视觉指示器620是发光二极管(LED),而

音频换能器625是扬声器。这些设备可以直接地耦合至电源670,以便当被激活时,即使处理器660和其他组件可能关闭以节省电池电量,这些设备也能保持通电并持续由通知机制所指示的一段时间。LED可以被编程为无限期地保持通电,直到用户采取动作来指示设备的通电状态为止。音频接口674用于向用户提供可听见的信号并从用户接收可听见的信号。例如,除了耦合到音频换能器625之外,音频接口674还可以被耦合至麦克风以接收可听见的输入,例如,以促进电话会话。根据本公开的实施例,麦克风还可以充当用于促进对通知的控制的音频传感器,如将在下文中所描述的。系统602还包括视频接口676,其使得板载相机630的操作能够记录静止图像、视频流等。

[0066] 实现系统602的移动计算设备600具有另外的特征或功能。例如,移动计算设备600还可以包括另外的数据存储设备(可移动的和/或不可移动的),例如磁盘、光盘、或磁带。在图6B中由非易失性存储区域668示出了这样另外的存储单元。

[0067] 如在上文中所描述的,将由移动计算设备600所生成或捕获的数据/信息以及经由系统602所存储的数据/信息本地地存储在移动计算设备600上,或者将数据存储可以在经由无线电接口层672或经由移动计算设备600和与移动计算设备600相关联的分别的计算设备(例如,分布式计算网络(例如,互联网)中的服务器计算机)之间的有线连接由设备来访问的任何数量的存储介质上。应当理解的是,可以经由移动计算设备600、经由无线电接口层672、或经由分布式计算网络来访问这样的数据/信息。类似地,根据公知的数据/信息传输和存储方式(包括电子邮件和协同数据/信息共享系统),这样的数据/信息可以容易地在计算设备之间传输以供存储和使用。

[0068] 图7图示了用于处理在计算系统处从远程源所接收的数据的系统的架构的一个方面,所述远程源例如个人计算机704、平板计算设备706、或移动计算设备708,如在上文所描述的。在服务器设备702处显示的内容可以被存储在不同的通信信道或其他存储类型中。例如,各种文档可以使用目录服务722、web门户724、邮箱服务726、即时消息传送存储单元728、或社交网站730来存储。请求处理程序组件721可以由与服务器设备702通信的客户端来实施,和/或页面生成组件720可以由服务器设备702来实施。服务器设备702可以通过网络715向诸如个人计算机704、平板计算设备706、和/或移动计算设备708(例如,智能电话)之类的客户端计算设备提供数据,或者从它们接收数据。作为示例,上文所描述的计算机系统可以以个人计算机704、平板计算设备706、和/或移动计算设备708(例如,智能电话)来实施。除了接收可用于在图形发起系统处预先处理或者在接收计算系统处进行后期处理的图形数据之外,计算设备的任何这些实施例都可以从存储单元716获取内容。

[0069] 图8图示了可以执行在本文中所公开的一个或多个方面的示例性平板计算设备800。另外,在本文中所描述的方面和功能可以在分布式系统(例如,基于云的计算系统)上操作,其中,应用功能、存储器、数据存储和获取以及各种处理功能可以通过诸如互联网或内联网之类的分布式计算网络而彼此远离地操作。各种类型的用户界面和信息可以经由板载计算设备显示器或者经由与一个或多个计算设备相关联的远程显示单元来显示。例如,各种类型的用户界面和信息可以在各种类型的用户界面和信息被投射于其上的墙壁表面上进行显示和交互。与本发明的方面可以利用其进行实践的许多计算系统的交互包括键击输入、触摸屏输入、语音或其他音频输入、手势输入——其中相关联的计算设备装备有用于捕捉和解译用于控制计算设备的功能的用户手势的检测(例如,相机)功能,等等。

[0070] 如将从前述公开所理解的,该技术的一个方面涉及一种系统,包括:至少一个处理器;以及存储指令的存储器,所述指令当由所述至少一个处理器执行时,实行一组操作。所述一组操作包括:在具有请求处理程序的基于web的组件中缓存包括一个或多个web资源的模板页面;由混合应用接收针对资源的请求;由所述请求处理程序来确定针对所述资源的所述请求是否能够使用所述基于web的组件中的所述模板页面被处理;当确定所述请求能够被处理时:请求所述资源;响应于所述请求,接收与所述资源相关联的内容;使用所接收到的内容来更新所述模板页面;以及由所述混合应用中的所述基于web的组件,使用由所述基于web的组件所缓存的所述一个或多个web资源中的至少一个web资源来显示经更新的模板页面。在示例中,确定所述请求是否能够被处理包括使用模式匹配来评估针对所述资源的所述请求。在另一个示例中,更新所述模板页面包括基于所接收的内容的至少一部分来执行对所述模板页面的原位更新。在另外的示例中,与所述资源相关联的所接收的内容包括所请求的资源的一部分。在另一个示例中,与所述资源相关联的内容包括与所述资源相关联的一个或多个另外的web资源的清单。在另外的示例中,更新所述模板页面还包括访问一个或多个另外的web资源中的至少一个,并且其中,显示经更新的模板页面还包括使用所述一个或多个另外的web资源中的至少一个。在另一个示例中,所述一组操作还包括:当确定所述请求不能够使用所述模板页面被处理时,在所述基于web的组件中加载所请求的资源。在另外的示例中,缓存所述模板页面包括基于所述模板页面来准备文档对象模型。在另一示例中,更新所述模板页面包括利用所接收的内容的至少一部分来至少更新所述文档对象模型。

[0071] 在另一方面中,该技术涉及一种计算机实现的方法。所述计算机实现的方法包括:在具有请求处理程序的基于web的组件中缓存与web应用相关联的模板页面,其中,所述模板页面包括一个或多个web资源;由混合应用来接收针对资源的请求;由所述请求处理程序来确定所述资源是否与所述web应用相关联以及所述模板页面是否可用于处理所述请求;基于确定所述资源与所述web应用相关联以及所述请求能够被处理:请求所述资源;响应于所述请求来接收与所述资源相关联的内容;基于所接收的内容来更新所述模板页面;以及由所述混合应用中的所述基于web的组件,使用由所述基于web的组件所缓存的所述一个或多个web资源中的至少一个web资源来显示经更新的模板页面;以及基于确定所述请求不能够使用所述模板页面被处理,在所述基于web的组件中加载所请求的资源。在示例中,更新所述模板页面包括基于所接收的内容的至少一部分来执行对所述模板页面的原位更新。在另一示例中,与所述资源相关联的内容包括与所述资源相关联的一个或多个另外的web资源的清单。在另外的示例中,更新所述模板页面还包括访问一个或多个另外的web资源中的至少一个web资源,并且其中,显示经更新的模板页面还包括使用一个或多个另外的web资源中的至少一个web资源。在另一示例中,缓存所述模板页面包括基于所述模板页面来准备文档对象模型。在另外的示例中,更新所述模板页面包括利用所接收内容的至少一部分来至少更新所述文档对象模型。

[0072] 在另外的方面,该技术涉及另一种计算机实现的方法。所述计算机实现的方法包括:在具有请求处理程序的基于web的组件中缓存包括一个或多个web资源的模板页面;由混合应用来接收针对资源的请求;由所述请求处理程序来确定针对所述资源的所述请求能够使用所述基于web的组件中的所述模板页面被处理;基于确定所述请求能够被处理:请求

所述资源；响应于所述请求来接收与所述资源相关联的内容；使用所接收的内容来更新所述模板页面；以及由所述混合应用中的所述基于web的组件，使用由所述基于web的组件所缓存的所述一个或多个web资源中的至少一个web资源来显示经更新的模板页面。在示例中，确定所述请求能够被处理包括确定所述模板页面和所请求的资源是否与web应用相关联。在另一示例中，缓存所述模板页面包括基于所述模板页面来准备文档对象模型。在另外的示例中，更新所述模板页面包括利用所接收的内容的至少一部分来至少更新所述文档对象模型。在另一示例中，与所述资源相关联的内容包括与所述资源相关联的一个或多个另外的web资源的清单。

[0073] 例如，在上文中参考根据本公开的方面的方法、系统、和计算机程序产品的框图和/或操作图而描述了本公开的各方面。在方框中所记录的功能/操作可以以不同于在任何流程图所示出的顺序来进行。例如，取决于所涉及的功能/操作，连续地示出的两个方框可以实际上是基本同时执行的，或者方框可以有时以相反的顺序执行。

[0074] 在该申请中所提供的一个或多个示例的描述和说明不旨在以任何方式限制或约束所要求保护的公开的范围。在该申请中所提供的方面、示例、和细节被认为足以传达所有权，并且使得本领域技术人能够制造并使用所要求保护的公开的最优模式。所要求保护的公开不应该被解释为限于在该申请中所提供的任何方面、示例、或细节。无论是组合地还是单独地示出和描述，(结构上和方法上两者的)各种特征旨在被选择性地包括或省略，以产生具有特定组的特征的实施例。在已经提供了本申请的描述和说明之后，本领域技术人可以预想落入在该申请中所实施的一般的发明概念的更宽泛的方面的精神内的变型、修改和替代方面，而不脱离所要求保护的公开的更宽泛的范围。

100

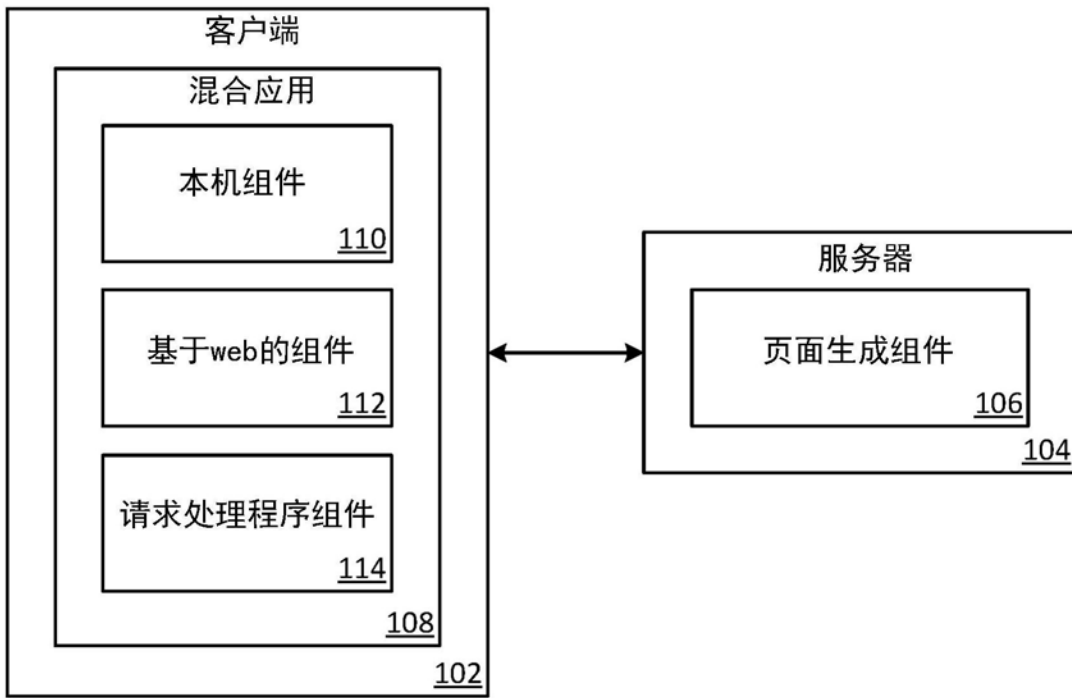


图1

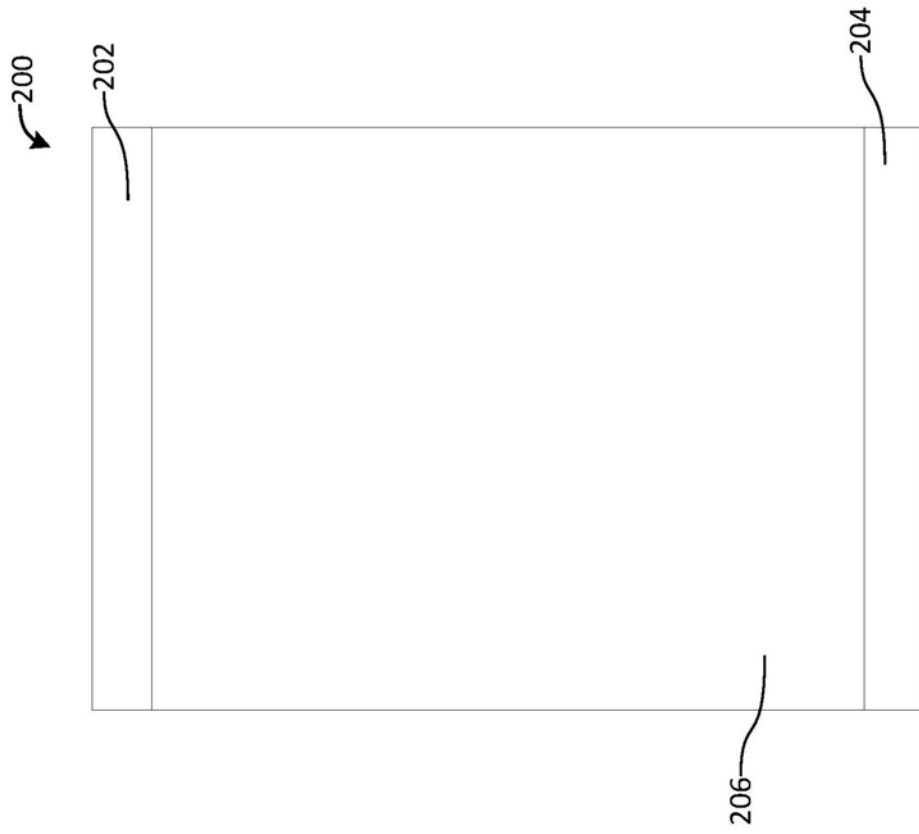


图2A

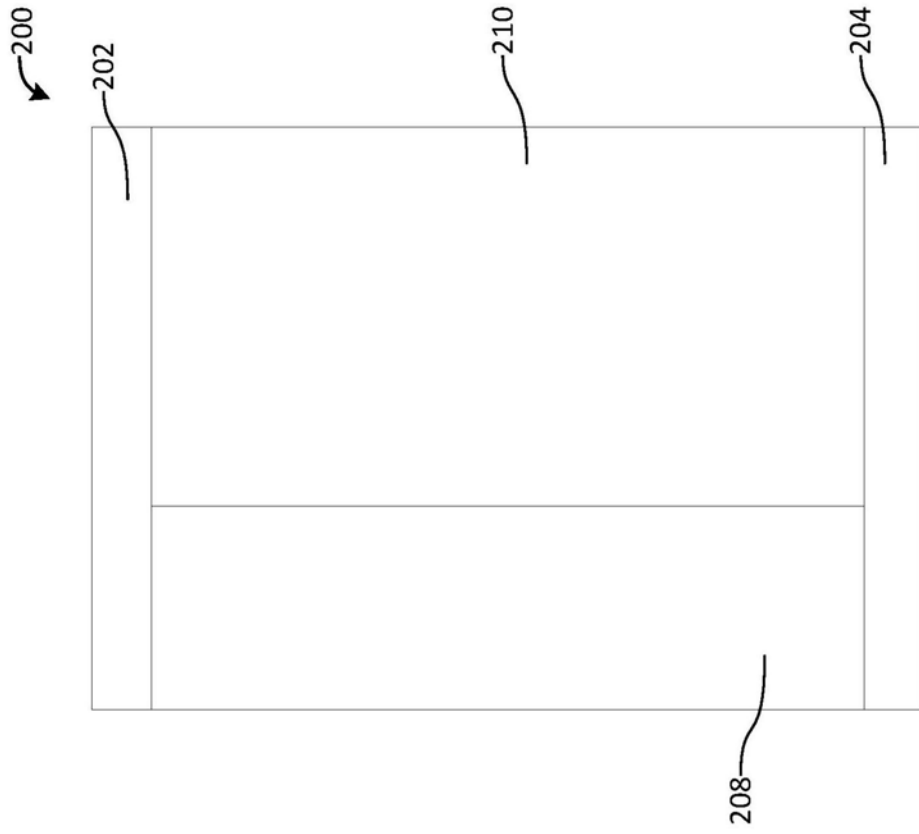


图2B

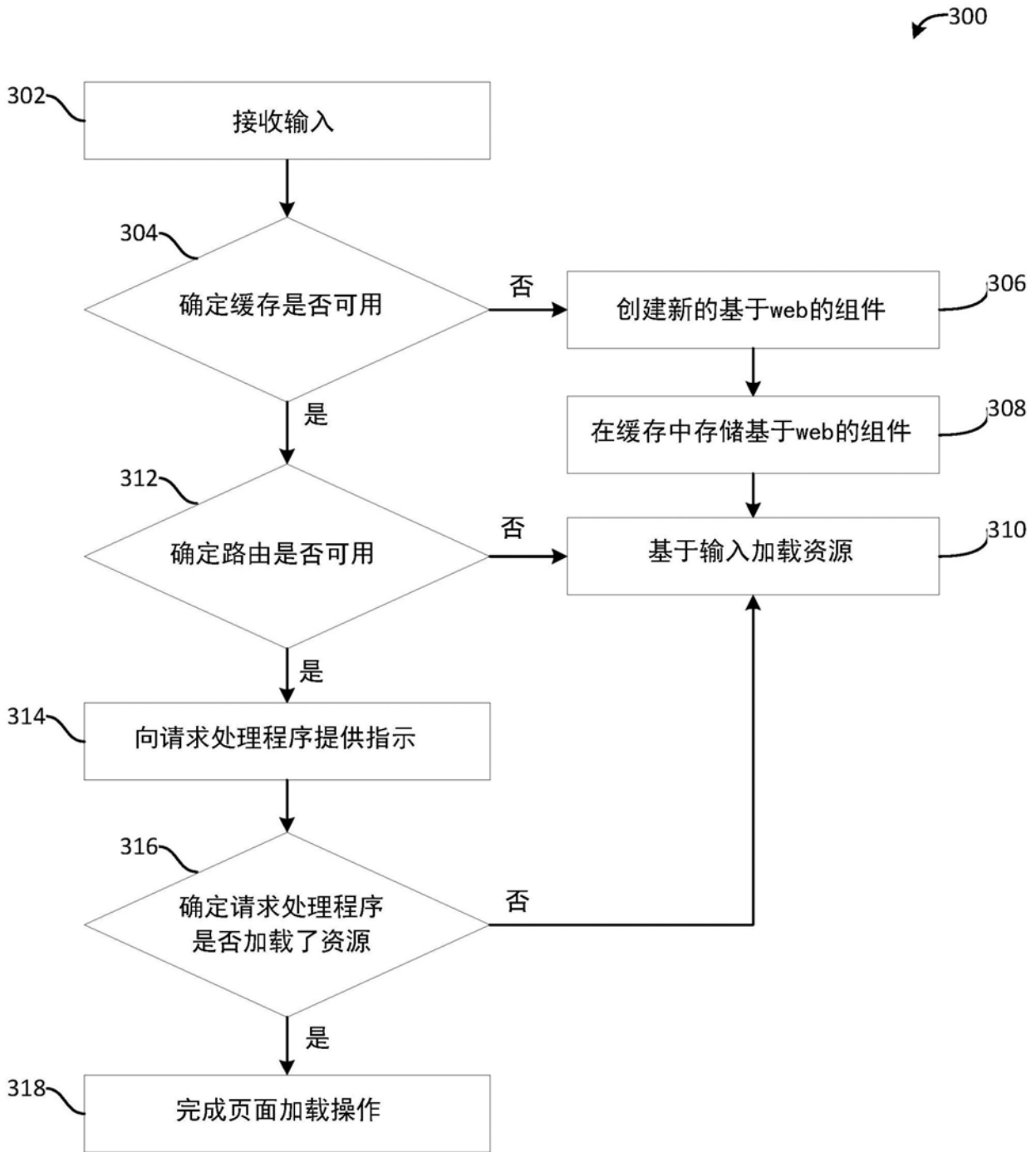


图3

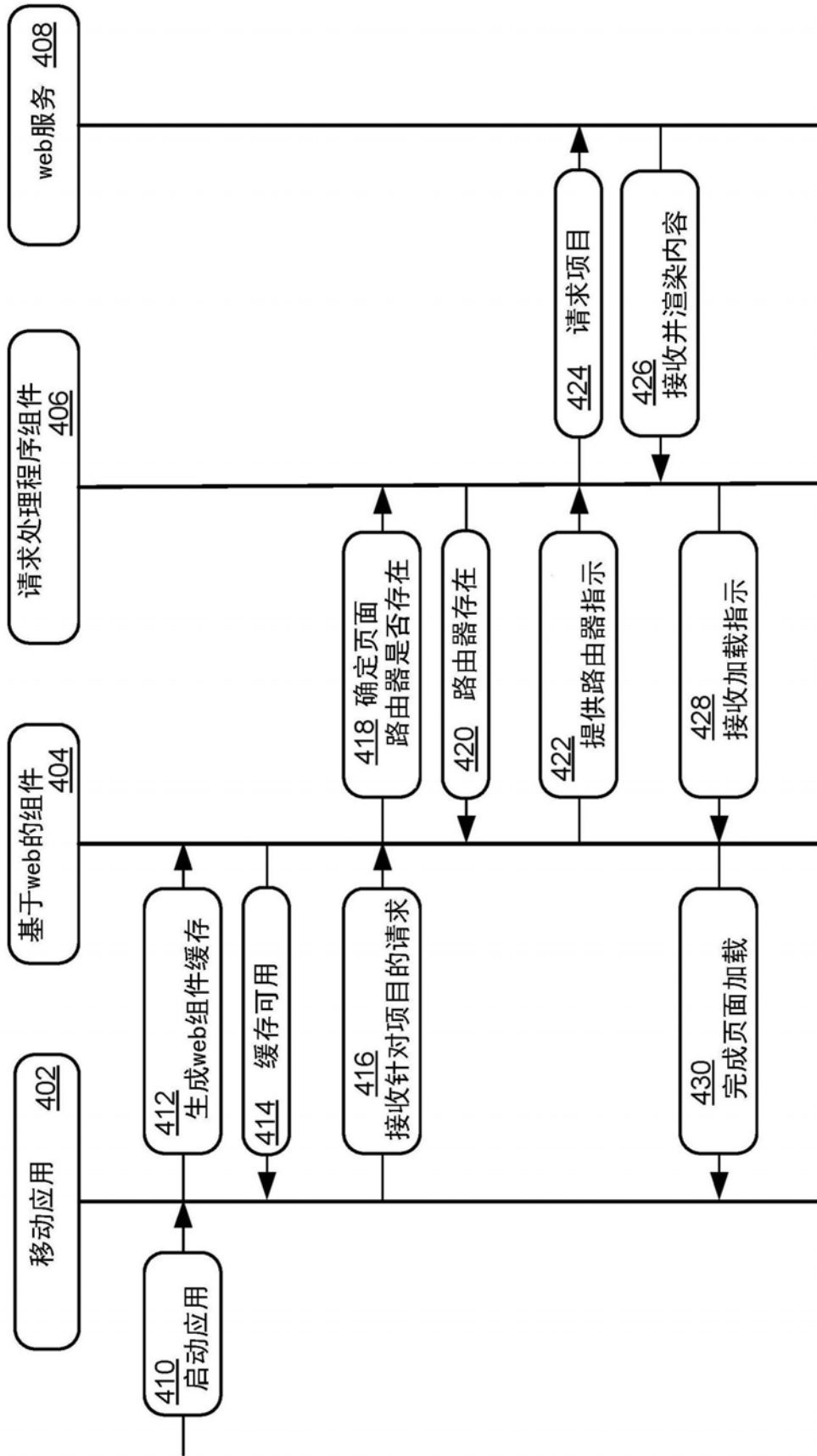


图4A

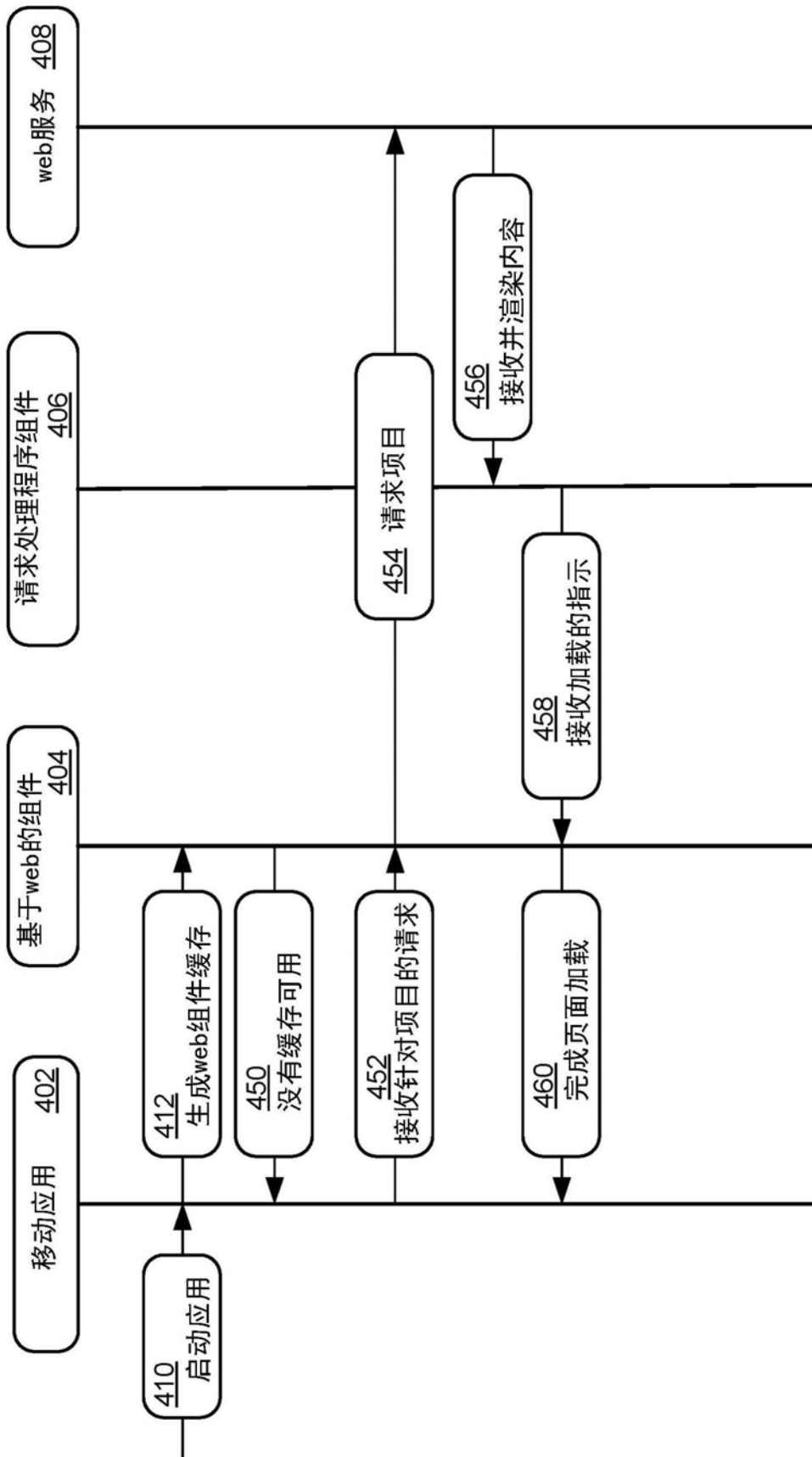


图4B

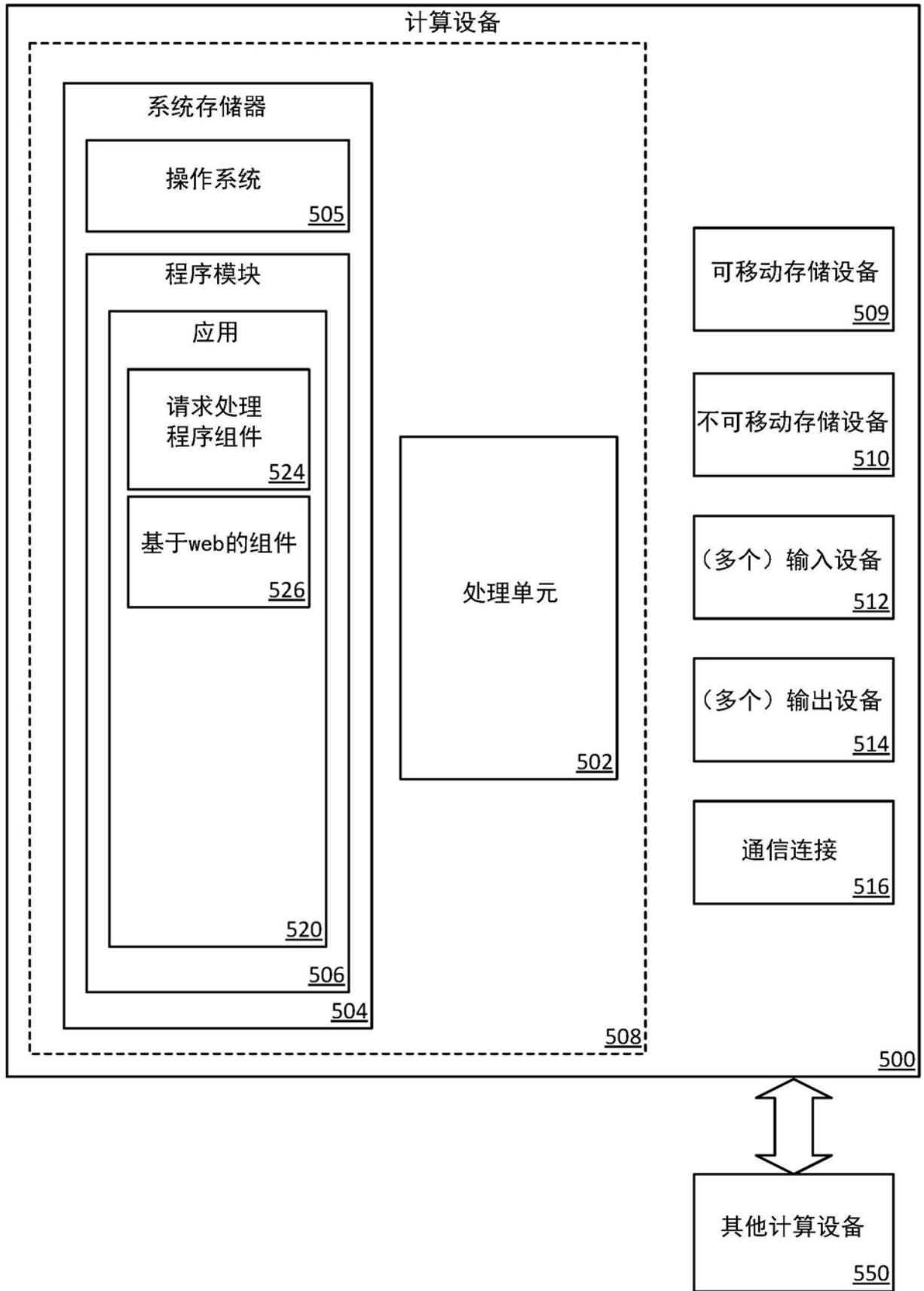


图5

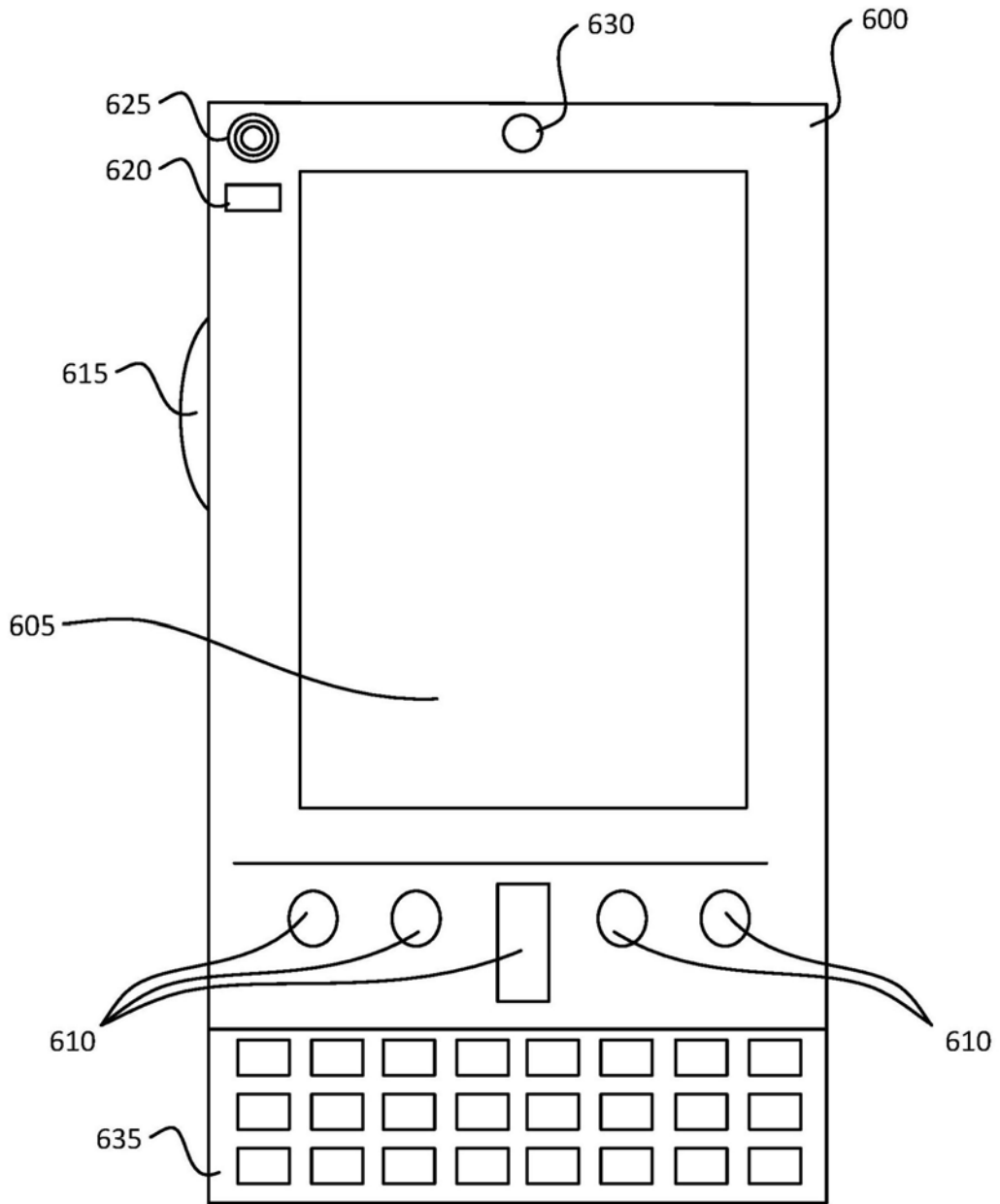


图6A

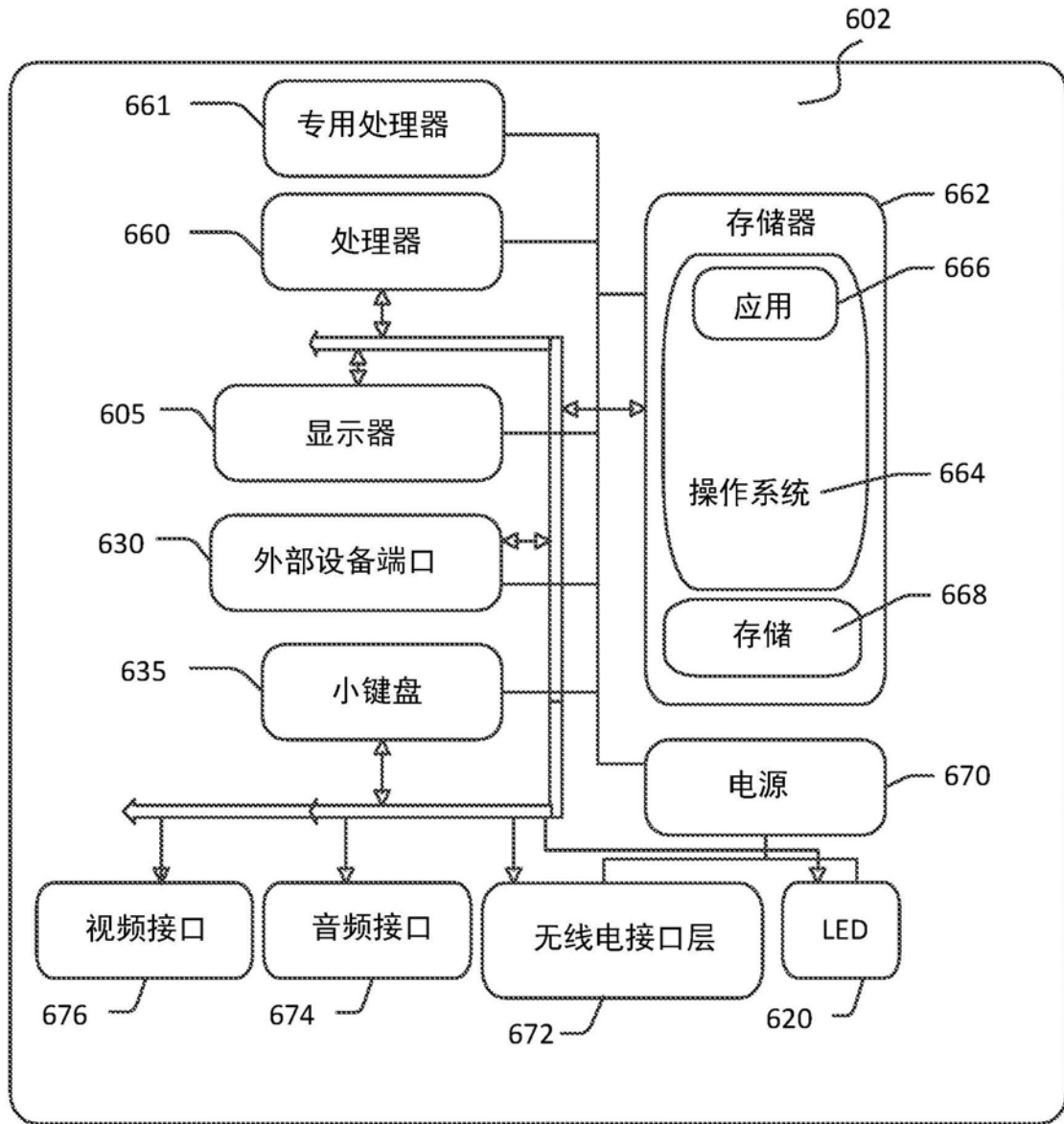


图6B

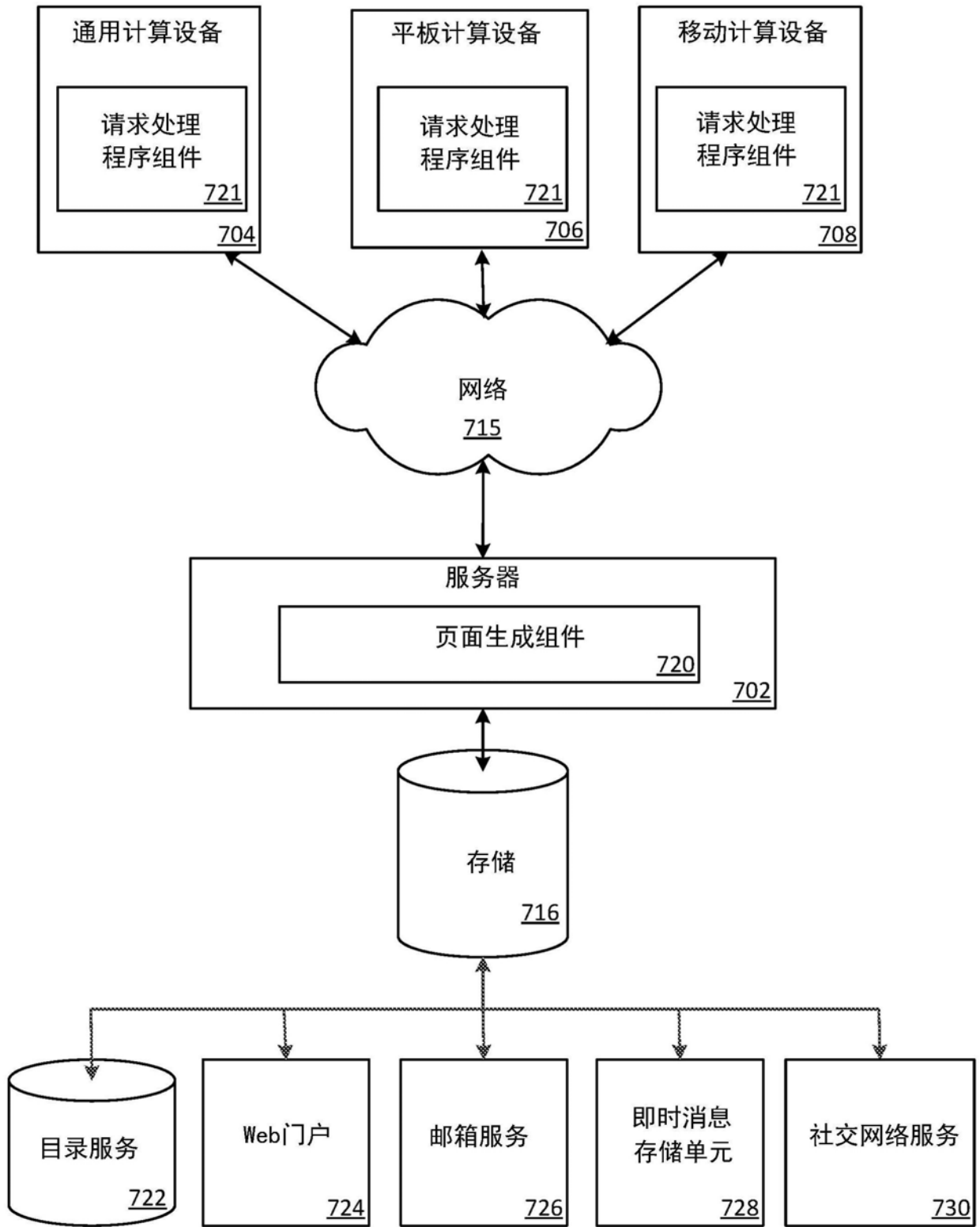


图7

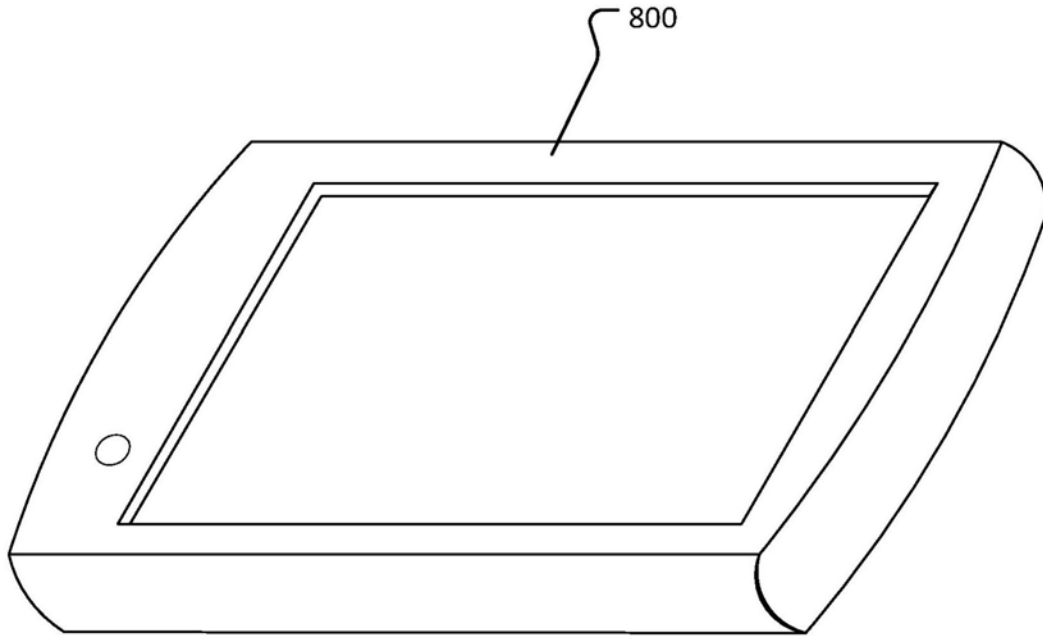


图8